

# **Software pro porovnání souborů a databázových objektů na serveru s lokálními zdroji**

Software for comparing files and database objects on the server with local resources

Jiří Kašpárek

---

Bakalářská práce  
2010



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Jiří KAŠPÁREK  
Osobní číslo: A07584  
Studijní program: B 3902 Inženýrská informatika  
Studijní obor: Informační a řídicí technologie

Téma práce: Software pro porovnání souborů a databázových objektů na serveru s lokálními zdroji

Zásady pro vypracování:

1. Analyzujte technologie Microsoft .NET Framework a Microsoft SQL Server.
2. Navrhněte způsob porovnání souborů umístěných na serveru se soubory v lokálním počítači.
3. Navrhněte způsob porovnání objektů v relační databázi na serveru s objekty v relační databázi na lokálním počítači.
4. Navrhněte uživatelské rozhraní aplikace pro zobrazení seznamu rozdílů mezi serverem a lokálním zdrojem.
5. Uvedený návrh softwarově realizujte.
6. Aplikaci realizujte na platformě Microsoft .NET Framework 3.5.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **MACDONALD, Matthew. ASP.NET 2.0 a C Sharp : tvorba dynamických stránek profesionálně. Brno : Zoner press, 2006. 1376 s. ISBN 80-86815-38-2.**
2. **NAGEL, Christian, EVJEN, Bill, GLYNN, Jay, WATSON, Karli, SKINNER, Morgan. C Sharp 2008 Programujeme profesionálně. Brno : Computer Press, 2009. 1904 s. ISBN 978-80-251-2401-7.**
3. **SHARP, John. Microsoft Visual C Sharp .Net krok za krokem. Brno : Mobil Media, 2002. 655 s. ISBN 80-86593-27-4.**
4. **PETZOLD, Charles. Programování Microsoft Windows Forms v jazyce C Sharp. Brno : Computer Press, 2006. 360 s. ISBN 80-251-1058-3.**
5. **SELLS, Chris. C Sharp a WinForms Programování formulářů Windows. Brno : Zoner press, 2005. 645 s. ISBN 80-86815-25-0.**
6. **BRUST, Andrew J. Mistrovství v programování SQL Serveru 2005. Brno : Computer Press, 2007. 847 s. ISBN 978-80-251-1607-4.**
7. **WALTERS, Robert E., COLES, Michael, RAE, Robert, FERRACCHIATI, Fabio, FARMER, Donald. Mistrovství v Microsoft SQL Server 2008. Brno : Computer Press, 2009. 864 s. ISBN 978-80-251-2329-4.**

Vedoucí bakalářské práce:

**Ing. Jiří Pálka**

Ústav elektroniky a měření

Datum zadání bakalářské práce:

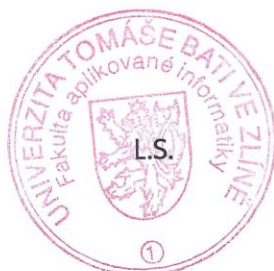
**5. března 2010**

Termín odevzdání bakalářské práce:

**1. června 2010**

Ve Zlíně dne 5. března 2010

prof. Ing. Vladimír Vašek, CSc.  
*děkan*



doc. Ing. Ivan Zelinka, Ph.D.  
*ředitel ústavu*



## **ABSTRAKT**

Hlavním cílem této bakalářské práce je vytvoření aplikace, která bude porovnávat soubory a objekty v relační databázi na serveru se soubory a objekty v relační databázi v lokálním počítači. V první části práce jsou popsány technologie a software, které byly použity při vytváření této aplikace. Dále je zde popsán způsob řešení porovnání jak pro soubory, tak pro objekty v relační databázi. Také je zde popis struktury a jednotlivých funkcí aplikace. Nakonec jsou zde náhledy jednotlivých oken aplikace s jejich popisem.

Klíčová slova: C Sharp, .NET, WinForms, SQL, porovnávání, soubory, databáze

## **ABSTRACT**

The main objective of this bachelor work is to create an application that will compare files and objects in a relational database on a server with files and objects in a relational database on a local computer. The first part describes technology and software used to create this application. Furthermore, there is described solution method for comparing files and objects in a relational database. There is also a description of the structure and functions of application. Finally, here are previews of application windows with their description.

Keywords: C Sharp, .NET, WinForms, SQL, comparison, files, database

Děkuji svému vedoucímu bakalářské práce Ing. Jiřímu Pálkovi za odbornou pomoc, cenné rady a čas, který mi věnoval.

**Prohlašuji, že**

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

**Prohlašuji,**

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....  
podpis diplomanta

**OBSAH**

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1 POUŽITÉ TECHNOLOGIE A SOFTWARE</b> .....	<b>11</b>
1.1 MICROSOFT .NET FRAMEWORK .....	11
1.2 C SHARP.....	12
1.3 WINDOWS FORMS.....	12
1.4 WEBOVÁ SLUŽBA .....	12
1.5 INTERNETOVÁ INFORMAČNÍ SLUŽBA .....	13
1.6 MESSAGE-DIGEST ALGORITHM 5 .....	13
1.7 TRIPLE DES .....	14
1.8 MICROSOFT VISUAL STUDIO.....	14
1.9 SQL .....	15
1.10 MICROSOFT SQL SERVER.....	15
1.10.1 Transact-SQL .....	16
<b>II PRAKTICKÁ ČÁST</b> .....	<b>17</b>
<b>2 NÁVRH POROVNÁVÁNÍ SOUBORŮ NA SERVERU SE SOUBORY V LOKÁLNÍM POČÍTAČI</b> .....	<b>18</b>
2.1 ZÍSKÁNÍ INFORMACÍ O SOUBORECH NA SERVERU .....	18
2.2 POROVNÁVÁNÍ INFORMACÍ O SOUBORECH.....	18
<b>3 NÁVRH POROVNÁVÁNÍ OBJEKTŮ V RELAČNÍ DATABÁZI NA SERVERU S OBJEKTY V RELAČNÍ DATABÁZI V LOKÁLNÍM POČÍTAČI</b> .....	<b>20</b>
3.1 ZÍSKÁNÍ INFORMACÍ O DATABÁZI.....	20
3.2 POROVNÁNÍ INFORMACÍ O TABULKÁCH V DATABÁZÍCH.....	20
3.3 POROVNÁNÍ INFORMACÍ O PROCEDURÁCH V DATABÁZÍCH .....	21
<b>4 NÁVRH GRAFICKÉHO UŽIVATELSKÉHO ROZHRANÍ</b> .....	<b>22</b>
<b>5 WEBOVÁ SLUŽBA</b> .....	<b>23</b>
<b>6 APLIKACE</b> .....	<b>24</b>

6.1	MAINFORM .....	24
6.2	DBCOMPARATORFORM .....	25
6.3	FILECOMPARATORFORM .....	26
6.4	COMPARERESULTSFORM .....	27
6.5	TEXTVIEWFORM .....	28
6.6	TABLEVIEWFORM .....	29
<b>7</b>	<b>PRÁCE S APLIKACÍ .....</b>	<b>30</b>
7.1	NASTAVENÍ NEZBYTNÁ PRO SPRÁVNÝ CHOD APLIKACE .....	30
7.2	NASTAVENÍ NEZBYTNÁ PRO SPRÁVNÝ CHOD WEBOVÉ SLUŽBY .....	30
7.3	POROVNÁNÍ SOUBORŮ .....	30
7.4	POROVNÁNÍ OBJEKTŮ V RELAČNÍ DATABÁZI .....	31
	<b>ZÁVĚR .....</b>	<b>32</b>
	<b>ZÁVĚR V ANGLIČTINĚ .....</b>	<b>33</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>34</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>35</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>37</b>
	<b>SEZNAM TABULEK .....</b>	<b>38</b>



## ÚVOD

Hlavním cílem této bakalářské práce je vytvoření aplikace, která bude porovnávat soubory a objekty v relační databázi na serveru se soubory a objekty v relační databázi v lokálním počítači. Cílem této aplikace je usnadnit vyhledání rozdílů ve webové aplikaci a k ní příslušné relační databázi. To může být užitečné například v situaci, kdy je v týmu vývojářů vyvíjena aplikace obsahující velké množství souborů nebo objektů v relační databázi a zároveň je nutné, aby tato aplikace nepřetržitě běžela na serveru. Potom při nahrávání oprav a nových částí aplikace potřebujeme zjistit, co vše se nám změnilo, jak v souborech, tak i v databázi. Pokud budeme přesně vědět, které soubory nebo objekty v relační databázi přibyly, které ubyly a které byly pozměněny, můžeme tím předejít různým chybám a s tím spojeným ztrátám způsobených nefunkčností aplikace.

Aplikace je realizována na platformě Microsoft .NET Framework 3.5. Tím pádem je aplikace kompatibilní s operačními systémy Microsoft Windows a bude moci komunikovat se servery založenými na Microsoft Windows Server a databázovými servery založenými na Microsoft SQL Server. Grafické uživatelské rozhraní je realizováno pomocí API Windows Forms.

Při návrhu algoritmu pro porovnání souborů a objektů v relační databázi na serveru se soubory a objekty v relační databázi v lokálním počítači bylo hlavním cílem dosáhnout co nejvyšší rychlosti zpracování a zároveň přenášet co nejmenší objemy dat mezi lokálním počítačem a serverem. Při návrhu grafického uživatelského rozhraní byl kladen důraz na to, aby byla aplikace uživatelsky přívětivá a přehledná.

## **I. TEORETICKÁ ČÁST**

# 1 POUŽITÉ TECHNOLOGIE A SOFTWARE

## 1.1 Microsoft .NET Framework

Microsoft .NET Framework je softwarový balík, který může být instalován na počítačích s operačním systémem Microsoft Windows. Obsahuje velké množství knihoven s hotovými řešeními pro běžné programátorské problémy a také prostředí pro běh aplikací vytvořených Microsoft .NET Framework.

.NET Framework nabízí širokou škálu funkcí včetně vytváření uživatelského rozhraní, přístup k datům, připojení k databázím, šifrování, vývoj webových aplikací, numerické algoritmy a komunikace v síti.

Programy napsané pro .NET Framework lze spustit v runtime prostředí známé jako Common Language Runtime. CLR poskytuje aplikaci abstrakci virtuálního stroje, takže programátor nemusí brát v úvahu možnosti konkrétního CPU, na kterém bude program spuštěn. CLR také poskytuje další důležité služby jako je bezpečnost, správa paměti a zpracování výjimek.

Nejpoužívanější programovací jazyky pro vývoj .NET aplikací jsou C Sharp, Visual Basic .NET a Delphi. K dispozici je nicméně řada dalších programovacích jazyků. Rodina .NET Framework zahrnuje také dvě verze pro vývoj pro mobilní telefony nebo embedded zařízení. [1]

Verze Microsoft .NET Framework:

- 1.0 – uvedena v roce 2002, přináší vývojové prostředí Visual Studio .net a programovací jazyk C Sharp 1.0
- 1.1 – uvedena v roce 2003 a přináší vývojové prostředí Visual Studio 2003
- 2.0 – uvedena v roce 2005, přináší nové verze jazyků C Sharp 2.0 a Visual Basic .NET 8.0 a vývojové prostředí Visual Studio 2005
- 3.0 – uvedena v roce 2007 a je pouze nadstavbou nad verzí 2.0
- 3.5 – uvedena v roce 2007, také pouze nadstavba nad verzí 2.0, přináší nové verze jazyků C Sharp 3.0 a Visual Basic .NET 9.0 a vývojové prostředí Visual Studio 2008
- 4.0 – uvedena v roce 2010, přináší nové verze jazyků C Sharp 4.0, Visual Basic .NET 10.0 a vývojové prostředí Visual Studio 2010

## 1.2 C Sharp

C Sharp je vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý společností Microsoft společně s platformou Microsoft .NET Framework. Je založen na jazycích C++, Java a Visual Basic 6.0. C Sharp lze využít k tvorbě databázových programů, webových aplikací, webových služeb, formulářových aplikací ve Windows, aplikací pro mobilní zařízení atd.

Je založen na Common Language Infrastructure, což je otevřená specifikace pro spustitelný kód a runtime prostředí, které tvoří jádro Microsoft .NET Framework. CLI instrukce jsou pomocí Just-In-Time kompilátoru překládány do instrukcí specifických pro danou platformu. JIT kompilace překládá část kódu programu do nativního strojového kódu až za běhu programu. C Sharp také používá unifikovaný typový systém zvaný Common Type System, který je nutný kvůli kompatibilitě .NET Frameworku s různými programovacími jazyky. Jazyk je také typově bezpečnější než C++. [1]

## 1.3 Windows Forms

Windows Forms je grafické API, které je součástí Microsoft .NET Framework. Poskytuje přístup ze spravovaného kódu k nativnímu rozhraní operačního systému Windows pomocí zabalení součástí stávajícího Windows API. Výsledné aplikace jsou řízené událostmi, které jsou aplikaci předávány od operačního systému. API poskytuje objekty pro práci s okny a také grafické objekty, jako jsou tlačítka, textboxy, menu, lišty se záložkami a další, které jsou běžnou součástí oken ve Windows. [2]

## 1.4 Webová služba

Webová služba je softwarový systém umožňující interakci dvou strojů na síti. Je popsána popisem ve strojově zpracovatelném formátu tzv. WSDL, který popisuje, jaké funkce služba nabízí a jak se na tyto funkce zeptat. S webovou službou komunikují ostatní stroje způsobem předepsaným v protokolu SOAP. Pro přenos informací využívá SOAP tzv. tunelování firewallu a zabalí zprávy do jiného známého transportního protokolu např. HTTP nebo SMTP. WSDL a SOAP jsou provedeny v syntaxi jazyka XML. XML je značkovací jazyk určený především pro výměnu dat mezi aplikacemi a pro publikování

dokumentů, u kterých popisuje strukturu z hlediska věcného obsahu jednotlivých částí a nezabývá se vzhledem. [3]

## 1.5 Internetová Informační Služba

Internetová Informační Služba je webový server a sada dalších rozšiřujících modulů pro hosting různých webových aplikací. IIS byla vyvinuta firmou Microsoft a lze provozovat na operačních systémech Windows. Je postavena na modulárním modelu, což umožňuje, aby na serveru běžely pouze ty moduly, které jsou třeba. IIS obsahuje například moduly:

- HTTP modul – Používá se k obsluze úkolů spojených s HTTP požadavky.
- Bezpečnostní modul – Stará se o zabezpečení serveru, například specifikuje autentizační schémata, provádí URL autorizaci a filtruje požadavky.
- Content modul – Zpracovává požadavky na statické soubory, vrací výchozí stránku, pokud je třeba a vypisuje obsah adresářů.
- Kompresní modul – Provádí kompresi odpovědí, aplikuje Gzip kompresi převodu kódování odpovědí a kompresi statického obsahu.
- Caching modul – Ukládá zpracovávané informace v paměti na serveru a z vyrovnávací paměti poskytuje data následných žádostí o stejný zdroj.
- Logovací a diagnostický modul – Provádí diagnostiku serveru a zaznamenává různé události probíhající na serveru a ukládá je do logů.

Služba podporuje komunikační protokoly jako FTP, FTPS, SMTP, NNTP, HTTP a HTTPS. [3]

## 1.6 Message-Digest Algorithm 5

Message-Digest algorithm 5 patří do rodiny hashovacích funkcí, které vytváří ze vstupních dat výstup fixní délky (otisk). MD5 vytváří výstupní otisk o délce 128bitů. Jeho hlavní vlastností je, že malá změna na vstupu vede k velké změně na výstupu a k vytvoření zásadně odlišného otisku. MD5 se často používá pro ukládání hesel. Kryptografická odolnost MD5 je rovna  $2^{128}$ . Nevýhodou MD5 je, že z otisku nelze zpět získat původní data. [4]

## 1.7 Triple DES

Triple DES je bloková šifra založená na šifrování Data Encryption Standard, které se aplikuje třikrát a tak se zvyšuje její odolnost proti prolomení. Nejčastěji používaná varianta pracuje s klíčem o celkové délce 168 bitů. Kryptografická odolnost Triple DES je rovna  $2^{168}$ . Triple DES tak umožňuje nejen data zašifrovat, ale také zpět rozšifrovat a získat tím původní data. [5]

## 1.8 Microsoft Visual Studio

Microsoft Visual Studio je integrované vývojové prostředí od společnosti Microsoft. Může být použito k vývoji konzolových aplikací, Windows Forms aplikací, webových stránek, webových aplikací a webových služeb. Aplikace je možno psát jak ve strojovém kódu, tak ve spravovaném kódu.

MS Visual Studio podporuje platformy Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework a Microsoft Silverlight. Také má vestavěné jazyky C/C++, Visual Basic .NET a C Sharp. Podpora dalších jazyků jako Chrome, F Sharp, Python a Ruby spolu s ostatními může být přidána jazykovými službami, které se dají nainstalovat zvlášť. Dále je podporováno XML/XSLT, HTML/XHTML, JavaScript a CSS. Existují i verze Visual Studia jen pro určitý jazyk, které uživatelům poskytují omezenější jazykové služby.

MS Visual Studio obsahuje editor kódu podporující IntelliSense a refaktorování. IntelliSense je nástroj pro automatické dokončování symbolických jmen. Slouží jako dokumentace a zároveň jako rozcestník pro názvy proměnných, funkcí a metod.

Debugger integrovaný v MS Visual Studiu pracuje jak na úrovni spravovaného kódu, tak na úrovni strojového kódu. Může být použit pro debugování aplikací napsaných ve kterémkoliv jazyce podporovaném Visual Studiem. Debugger podporuje breakpointy, které umožňují zastavit běh programu na určité pozici a také watch, který sleduje hodnoty proměnných za běhu procesu. Breakpointy mohou být podmíněné. Může také vstupovat do funkcí, aby je debugoval uvnitř, nebo je přejít. Také dovoluje upravovat funkce za běhu a poté pokračovat v debugování.

Další vestavěné nástroje zahrnují designér formulářů pro tvorbu GUI aplikací, designér webových stránek, tříd nebo databázových schémat. Také je možné rozšíření přidávat, což umožňuje vylepšit funkčnost na téměř každé úrovni. [1]

## 1.9 SQL

SQL je standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích. Původně založen na relační algebře. Poskytuje příkazy pro manipulaci s daty, definici dat, řízení přístupových práv, řízení transakcí a další speciální příkazy.

SQL byl vyvinut na počátku 70. let 20. století firmou IBM pod názvem SEQUEL. V roce 1981 pak společnost IBM uvedla novou verzi jazyka a jazyk byl přejmenován na SQL. V dalších letech pak byly vydány standardy SQL-86, SQL-92 a SQL-99, které byly pojmenovány vždy podle roku jejich přijetí.

Jazyk SQL je rozdělen do několika jazykových prvků:

- Doložky, které nejsou v některých případech povinné a tvoří základ dotazů.
- Výrazy, které mohou produkovat buď skalární hodnoty, nebo tabulky skládající se ze sloupců a řádků dat.
- Predikáty, které vytvářejí podmínky a používají se k omezení dotazů.
- Dotazy, načítají data na základě specifických kritérií.
- Prohlášení, které ovlivňují schémata, transakce, tok programu, spojení nebo diagnostiku. Měly by obsahovat ";" jako ukončovač, ale není to vyžadováno na všech platformách. [6]

## 1.10 Microsoft SQL Server

Microsoft SQL Server je relační databázový systém vytvořený firmou Microsoft. Jeho hlavními dotazovacími jazyky jsou SQL a T-SQL. Server je dostupný v několika odlišných edicích, které mají různou skladbu funkcí podle zaměření na různé uživatele.

MS SQL Server zahrnuje následující nástroje:

- Relační databáze – Relační databázový stroj s vysokým výkonem a podporou strukturovaných a nestrukturovaných dat (XML).

- Služba Replication Services – Zajišťuje replikaci dat pro aplikace zpracovávající distribuovaná či mobilní data.
- Služba Notification Services – Zajišťuje zasílání upozornění pro vývoj a nasazení škálovatelných aplikací, které mohou na připojená i mobilní zařízení zasílat individuální požadavky.
- Služba Integration Services – Zajišťuje funkce extrakce, transformace a načítání dat pro datové sklady a integraci dat.
- Služba Analysis Services – Zajišťuje OLAP pro rychlou a pokročilou analýzu velkých a složitých datových sad s využitím vícedimenzionálních úložišť.
- Služba Reporting Services – Zajišťuje vytváření, správu a zasílání tradičních papírových i interaktivních webových sestav.
- Nástroje pro správu – Nabízí integrované nástroje pro pokročilou správu a ladění databází a umožňuje úzkou integraci s nástroji, jako jsou například Microsoft Operations Manager a Microsoft Systems Management Server.
- Nástroje pro vývojáře – Nabízí integrované nástroje pro databázový stroj, extrakci, transformaci a načítání dat, dolování dat, funkce OLAP a vytváření sestav, které jsou úzce integrovány se sadou Microsoft Visual Studio a poskytují komplexní funkce pro vývoj aplikací. Jednotlivé součásti MS SQL Serveru jsou pak dodávány s vlastními objektovými modely a sadou aplikačních programovacích rozhraní.

### 1.10.1 Transact-SQL

Transact-SQL je proprietární rozšíření pro SQL od společností Microsoft a Sybase. Všechny aplikace komunikují s instancí MS SQL Serveru pomocí T-SQL bez ohledu na uživatelské rozhraní aplikace. T-SQL rozšiřuje SQL některými doplňkovými funkcemi:

- Přidává Control-of-Flow jazyk – klíčová slova pro řízení toku programu.
- Přidává lokální proměnné.
- Přidává různé podpůrné funkce pro zpracování řetězců, datumů, matematiku, atd.
- Pozměňuje „DELETE“ a „UPDATE“. [6]



## II. PRAKTICKÁ ČÁST

## 2 NÁVRH POROVNÁVÁNÍ SOUBORŮ NA SERVERU SE SOUBORY V LOKÁLNÍM POČÍTAČI

### 2.1 Získání informací o souborech na serveru

K získání informací o souborech na serveru je použita webová služba, která je umístěna na serveru, se kterým chceme naše lokální soubory porovnávat. Webová služba poskytuje funkci, která nám vrátí informace o souborech na serveru ve formě seznamu tabulek. Seznam obsahuje jednu tabulku, která obsahuje tři sloupce s daty. První obsahuje jméno souboru a relativní cestou k němu, druhý obsahuje velikost souboru v bytech a třetí obsahuje hash obsahu souboru. Aplikace si poté zavolá příslušnou funkci ve webové službě na zadané adrese, načte z ní informace o souborech a uloží si je lokálně.

### 2.2 Porovnávání informací o souborech

Porovnávání informací o souborech začíná opakováním cyklu, který běží tak dlouho, dokud je ve frontě uložena nějaká cesta k lokálnímu adresáři. Cyklus je inicializován adresářem, který byl uživatelem zadán jako kořenový pro porovnávání. Tento cyklus pak nejprve vyjme z fronty cestu k jednomu lokálnímu adresáři. Všechny adresáře v tomto adresáři uloží do fronty. Následně se pak snaží nalézt soubor se stejnou relativní cestou v informacích ze serveru a to pro každý soubor v tomto adresáři. Pokud nalezne shodu, porovná velikosti a hashe obou souborů, a pokud nejsou oba soubory v obou parametrech shodné, zapíše tento rozdíl do výstupní tabulky. Soubor je pak odstraněn z informací o souborech ze serveru. Když není nalezen soubor se stejnou relativní cestou na serveru, je zapsán do výstupní tabulky jako chybějící na serveru. Poté co doběhne tento cyklus, jsou soubory zbývající v informacích ze serveru přepsány do výstupní tabulky a označeny jako chybějící na lokálním počítači.

Aby bylo možné porovnávat obsahy souborů a zároveň nebylo třeba přenášet plný obsah souborů, je obsah každého souboru převeden na hash a ze serveru je již načítán pouze tento hash. Algoritmem MD5 je vytvořen hash o délce 128 bitů, což ve většině případů značně zmenší objem přenášených dat. Zároveň je ale algoritmem zaručeno, že i nepatrná změna v obsahu souboru se v hashi projeví jeho změnou. Také porovnání se tím zrychlí, protože porovnáваме pouze krátké řetězce. To vše má ale za následek potřebu většího množství procesorového času, kvůli nutnosti hashe vypočítat. Protože je dnes stále

síťový přenos dat limitním faktorem, ale výkonu mají dnešním moderní procesory velké množství, vyplatí se tuto optimalizaci implementovat.

Algoritmus také snižuje množství potřebných průchodů skrz informace ze serveru tím, že nalezené soubory z těchto informací odstraňuje, čímž snižuje množství dat, které se opakovaně prochází. Také při načítání informací ve webové službě algoritmus prochází adresářový strom na serveru a ukládá tyto informace ve stejném pořadí, v jakém se pak budou procházet lokální soubory. To ve spojení s odstraňováním již nalezených souborů má za následek, že algoritmus často najde lokální soubor na začátku informací ze serveru a zbytek již pak nebude třeba procházet.

### **3 NÁVRH POROVNÁVÁNÍ OBJEKTŮ V RELAČNÍ DATABÁZI NA SERVERU S OBJEKTY V RELAČNÍ DATABÁZI V LOKÁLNÍM POČÍTAČI**

#### **3.1 Získání informací o databázi**

Informace o databázi jsou získávány stejným způsobem jak ze serveru, tak z lokální databáze a není zde třeba využívat webovou službu. Aplikace se připojí k databázi za pomoci nástrojů poskytovaných .NET Frameworkem. Poté zavolá dotaz, který vrátí zpět potřebná data. V případě tabulky dotaz vrátí seznam tabulek, kde každá tabulka obsahuje určitou sadu informací o tabulce. Například to je seznam sloupců včetně informací o sloupcích nebo seznam cizích klíčů včetně informací. U procedur dotaz vrací text dané procedury.

#### **3.2 Porovnání informací o tabulkách v databázích**

Nejprve jsou vytvořeny seznamy seznamů informací o tabulce a to jak pro lokální, tak i pro serverovou databázi. Každý tento podseznam obsahuje kompletní sadu informací o dané tabulce a je pojmenován podle této tabulky, aby usnadnilo rozlišení informací o tabulce v seznamu.

Poté algoritmus pro každou tabulku v seznamu tabulek serverové databáze prochází seznam tabulek v lokální databázi a snaží se najít tabulku stejného jména. Pokud najde tabulku stejného jména, porovná postupně všechny položky v informacích o tabulkách a pokud najde rozdíl, zapíše tuto tabulku do výstupní tabulky jako rozdílnou. Poté tuto nalezenou tabulku odstraní ze seznamu tabulek v serverové databázi. Pokud tabulku se stejným jménem nenajde, zapíše ji do výstupní tabulky jako chybějící na serveru. Poté co projde všechny tabulky v seznamu tabulek v serverové databázi, jsou zbývající tabulky přepsány do výstupní tabulky a označeny jako chybějící v lokální databázi.

I zde algoritmus využívá optimalizace spojené s mazáním nalezených tabulek a stejným seřazením obou seznamů tabulek.

### 3.3 Porovnání informací o procedurách v databázích

Nejprve je vytvořena tabulka s informacemi o procedurách v serverové databázi. Řádek tabulky vždy obsahuje název procedury a k ní příslušný text procedury.

Poté algoritmus postupně načítá jména jednotlivých procedur v lokální databázi. Pro každou tuto proceduru následně hledá proceduru stejného jména v informacích o procedurách v serverové databázi. Pokud najde proceduru stejného jména, porovná texty obou procedur a pokud najde rozdíl, zapíše tento rozdíl do výstupní tabulky. Poté tuto nalezenou proceduru odstraní z informací o procedurách v serverové databázi. Pokud proceduru se stejným jménem nenajde, zapíše ji do výstupní tabulky jako chybějící na serveru. Poté co projde všechny procedury v informacích o procedurách v serverové databázi, jsou zbývající procedury přepsány do výstupní tabulky a označeny jako chybějící v lokální databázi.

Také zde algoritmus využívá optimalizace spojené s mazáním nalezených procedur a seřazením informací o procedurách v serverové databázi do stejného pořadí v jakém se pak budou procházet procedury v lokální databázi.

## 4 NÁVRH GRAFICKÉHO UŽIVATELSKÉHO ROZHŘANÍ

Grafické uživatelské rozhraní aplikace je rozděleno do několika oken. Hlavní okno, které se objeví po spuštění aplikace, je výchozím bodem pro většinu akcí prováděných uživatelem. Okno zobrazuje seznam vytvořených komparátorů a také ovládací prvky pro práci s nimi. Z hlavního okna může uživatel přejít na okno pro vytvoření nebo editaci souborového komparátoru. Zde zadá všechny údaje potřebné pro porovnání souborů na serveru se soubory v lokálním počítači. Pro databázový komparátor je zde velmi podobné okno, které se liší pouze sadou zadávaných údajů, které odpovídají potřebám pro porovnání objektů v relační databázi na serveru s objekty v relační databázi v lokálním počítači.

Z hlavního okna je také možné jednotlivé komparátory spustit. Po spuštění se zobrazí okno, které zobrazuje výsledky porovnání. Okno dále obsahuje ovládací prvky pro práci s těmito výsledky. Také zobrazuje postup při načítání a porovnávání souborů nebo objektů v relační databázi. Toto načítání a porovnávání pak běží na pozadí v jiném vlákne, než běží celé GUI aplikace. Tyto činnosti běží v jiném vlákne proto, aby GUI reagovalo na akce uživatele i při těchto časově náročných činnostech.

Z okna výsledků porovnání je možné si jednotlivé nalezené rozdílné soubory nebo objekty v relační databázi prohlížet. Toto prohlížení zajišťují dvě různá okna. Jedno pro prohlížení souboru nebo procedury v relační databázi v textové podobě. Druhé pro prohlížení informací o tabulce v relační databázi a to v podobě sady tabulek. I u těchto oken probíhá načítání na pozadí v jiném vlákne, aby GUI aplikace stále reagovalo.

Všechna tato okna jsou navržena z prvků, které jsou součástí API Windows Forms.

## 5 WEBOVÁ SLUŽBA

Webová služba je používána aplikací jako zdroj informací o souborech na serveru. K tomu má webová služba dvě veřejné webové metody. První z nich „GetFiles“ vrací seznam souborů v celém podstromu zadaného adresáře. Každý řádek tohoto seznamu obsahuje relativní cestu k souboru, velikost souboru v bytech a hash obsahu souboru. Této metodě je možné zadat, aby ignorovala některé soubory a adresáře. Druhou metodou je „GetFile“ a tato metoda vrací obsah zadaného souboru ve formě textového řetězce.

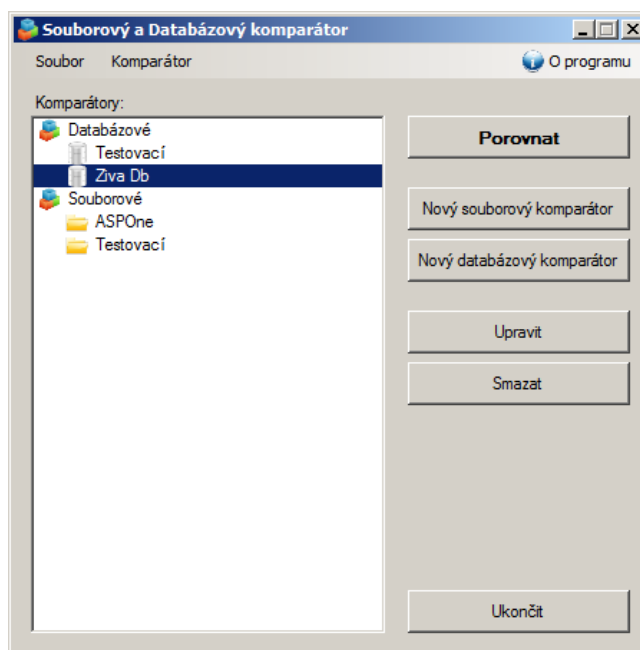
Přístup k oběma metodám je chráněn pomocí přístupového klíče. Přístupový klíč je společný pro celou webovou službu a administrátor tento klíč nastavuje v souboru „web.config“ příslušné webové služby. Při přenosu je klíč ještě navíc chráněn pomocí algoritmu MD5.

Webová služba také zachytává všechny výjimky vzniklé při jejím provozu. Tyto výjimky poté zaznamenává do souboru, aby na ně mohl administrátor později reagovat.

## 6 APLIKACE

### 6.1 MainForm

„MainForm“ je hlavním oknem aplikace (Obr. 1). Hned po startu aplikace je vytvořeno hlavní vlákno aplikace a v něm je v zápětí spuštěno a otevřeno právě toto okno. Okno má nahoře menu s funkcemi, pod ním je na levé straně seznam s komparátory a na pravé straně sloupec tlačítek pro práci s komparátory (Tab. 1).



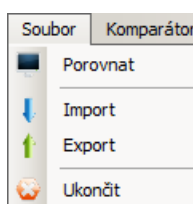
Obr. 1. Hlavní okno aplikace

Tab. 1. Funkce tlačítek hlavního okna

Jméno tlačítka	Funkce tlačítka
Porovnat	Spustí vybraný komparátor.
Nový souborový komparátor	Otevře okno na vytvoření nového souborového komparátoru.
Nový databázový komparátor	Otevře okno na vytvoření nového databázového komparátoru.
Upravit	Otevře okno s úpravami vybraného komparátoru.
Smazat	Smaže vybraný komparátor.
Ukončit	Ukončí celou aplikaci.

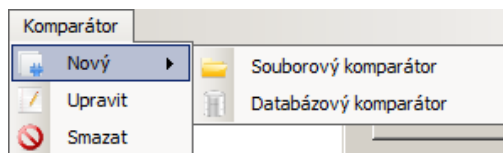


Položky v menu „Soubor“ (Obr. 2) označené jako „Porovnat“ a „Ukončit“ mají stejné funkce jako stejně pojmenovaná tlačítka hlavního okna. Položka „Import“ slouží k načtení seznamu komparátorů z XML souboru. Data tohoto souboru jsou navíc zašifrována algoritmem Triple DES. Toto šifrování se provádí proto, aby nebylo možné jednoduše přechíst přístupové klíče a přístupové údaje k databázím, které jsou uloženy v souboru. Položka „Export“ slouží k vytváření těchto zašifrovaných XML souborů se zálohami seznamu komparátorů.



*Obr. 2. Menu  
„Soubor“  
hlavního okna*

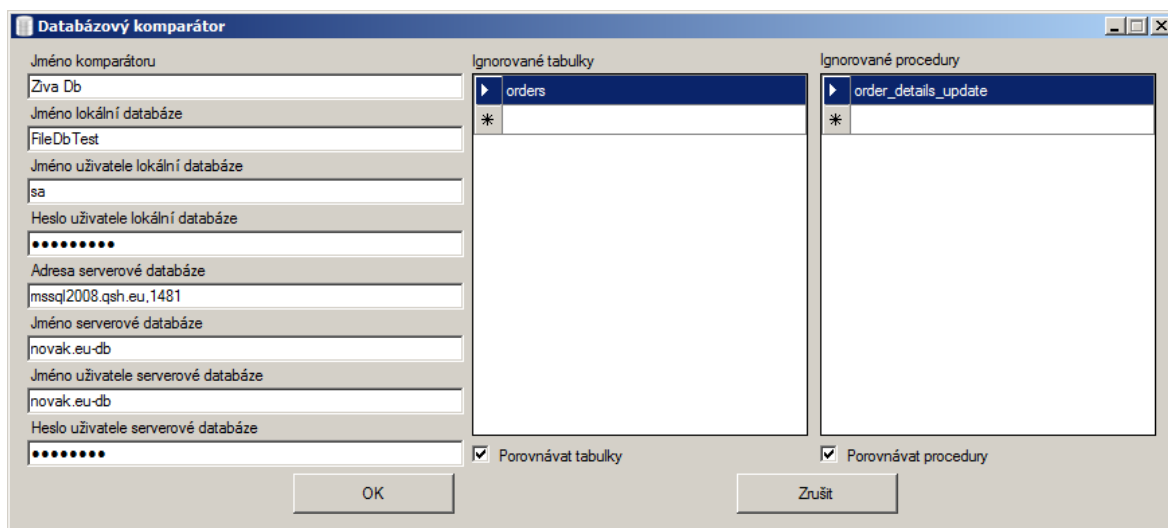
Položky v menu „Komparátor“ (Obr. 3) mají stejné funkce jako stejně pojmenovaná tlačítka hlavního okna.



*Obr. 3. Menu „Komparátor“  
hlavního okna*

## 6.2 DbComparatorForm

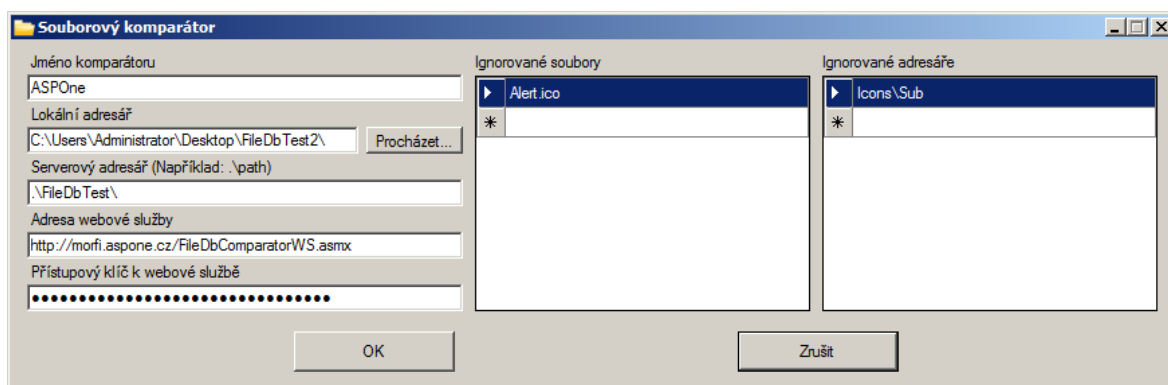
„DbComparatorForm“ je okno pro vytváření a úpravu databázových komparátorů (Obr. 4). Okno obsahuje v levé části textboxy pro zadání údajů potřebných pro porovnání objektů v relační databázi na serveru s objekty v relační databázi v lokálním počítači. V pravé části jsou dva seznamy pro zadání tabulek a procedur, které se mají při porovnávání ignorovat. Pod těmito seznamy jsou checkboxy, které umožňují vypnout nebo zapnout porovnávání tabulek a procedur. Tlačítko „OK“ slouží k uložení zadaných údajů a následnému zavření okna. Tlačítko „Zrušit“ slouží k zavření okna bez uložení údajů.



Obr. 4. Okno pro vytváření a úpravu databázových komparátorů

### 6.3 FileComparatorForm

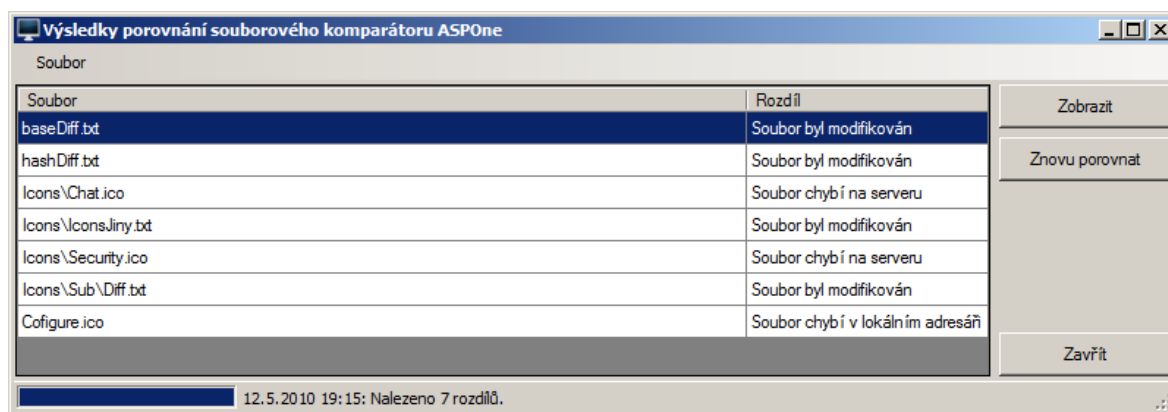
„FileComparatorForm“ je okno pro vytváření a úpravu souborových komparátorů (Obr. 5). Okno obsahuje v levé části textboxy pro zadání údajů potřebných pro porovnání souborů na serveru se soubory v lokálním počítači. Tlačítko „Procházet...“ vyvolává dialog pro zadání cesty k adresáři v lokálním počítači. V pravé části jsou dva seznamy pro zadání souborů a adresářů, které se mají při porovnávání ignorovat. Tlačítko „OK“ slouží k uložení zadaných údajů a následnému zavření okna. Tlačítko „Zrušit“ slouží k zavření okna bez uložení údajů.



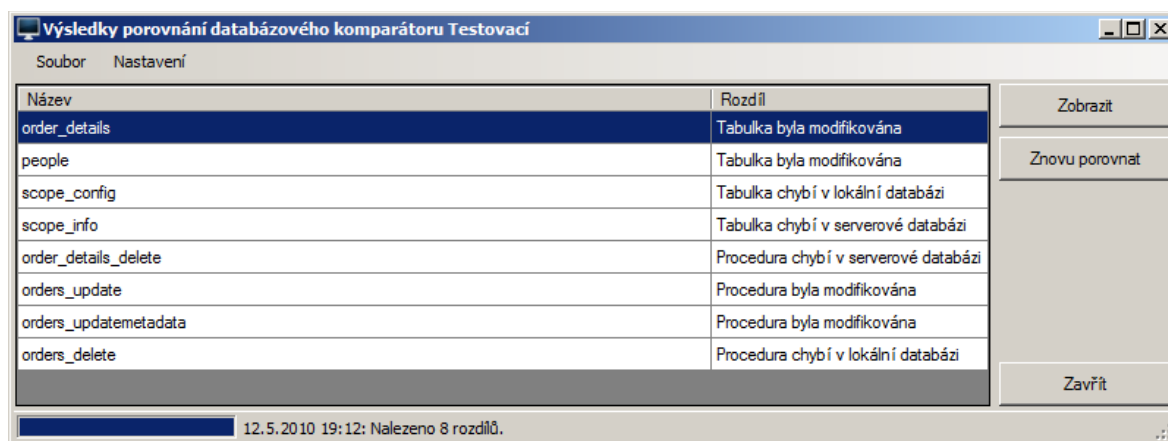
Obr. 5. Okno pro vytváření a úpravu souborových komparátorů

## 6.4 CompareResultsForm

„CompareResultsForm“ je okno s výsledky porovnávání daného komparátoru (Obr. 6-7). V levé části okna je tabulka s výsledky vyhledání komparátoru. Tabulka obsahuje název souboru nebo objektu v relační databázi a nalezený rozdíl. V pravé části okna je sloupec s tlačítky (Tab. 2).



Obr. 6. Okno výsledků porovnání pro souborový komparátor

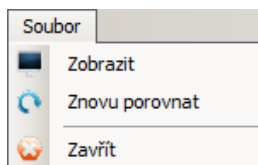


Obr. 7. Okno výsledků porovnání pro databázový komparátor

Tab. 2. Funkce tlačítek okna výsledků porovnání komparátoru

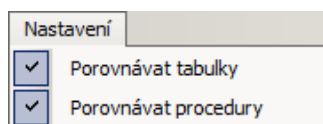
Jméno tlačítka	Funkce tlačítka
Zobrazit	Otevře okno, které zobrazí vybraný soubor nebo objekt v relační databázi.
Znovu porovnat	Spustí znovu porovnání, které zaktualizuje data v tabulce výsledků.
Zavřít	Zavře dané okno.

Položky v menu „Soubor“ (Obr. 8) mají stejné funkce jako stejně pojmenovaná tlačítka okna výsledků porovnání komparátoru.



*Obr. 8. Menu  
„Soubor“ okna  
výsledků porovnání  
komparátoru*

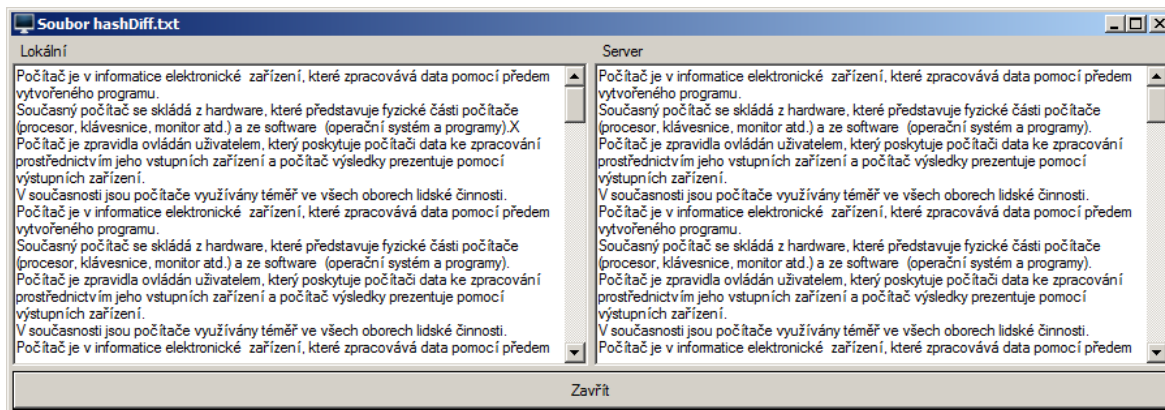
Menu „Nastavení“ (Obr. 9) umožňuje vypnout nebo zapnout porovnávání tabulek a procedur. Toto menu je dostupné pouze pokud okno zobrazuje výsledky pro databázový komparátor.



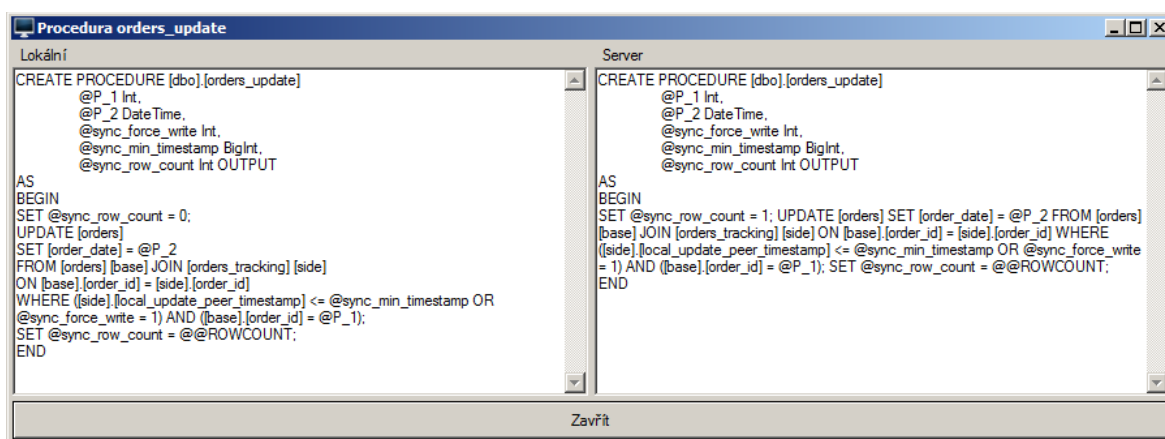
*Obr. 9. Menu  
„Nastavení“ okna  
výsledků porovnání  
komparátoru*

## 6.5 TextViewForm

„TextViewForm“ je okno, které zobrazuje zadaný soubor (Obr. 10) nebo proceduru (Obr. 11) v relační databázi v textové podobě. V levé části okna je zobrazen soubor nebo procedura v lokálním počítači a v pravé části soubor nebo procedura na serveru. Dole je tlačítko, které zavře dané okno.



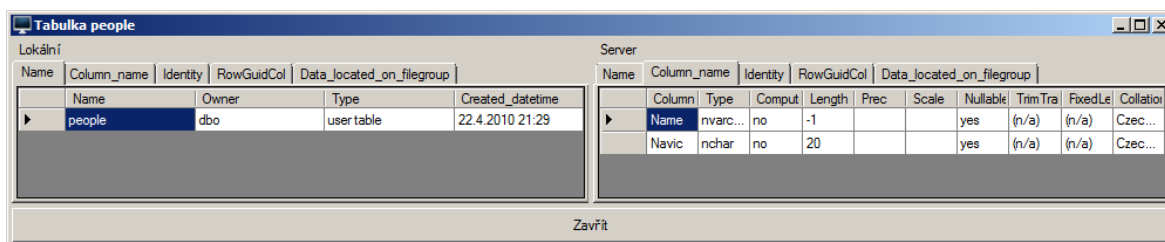
Obr. 10. Okno zobrazující soubor



Obr. 11. Okno zobrazující proceduru v relační databázi

## 6.6 TableViewForm

„TableViewForm“ je okno, které zobrazuje informace o zadané tabulce v relační databázi v podobě seznamu tabulek s informacemi (Obr. 12). V levé části okna jsou zobrazeny informace o tabulce v lokální databázi a v pravé části tabulka v databázi na serveru. Dole je tlačítko, které zavře dané okno. Seznam tabulek s informacemi je zobrazen pomocí sady záložek, kde každá záložka zobrazuje jednu tabulku s informacemi.



Obr. 12. Okno zobrazující informace o tabulce v relační databázi

## 7 PRÁCE S APLIKACÍ

### 7.1 Nastavení nezbytná pro správný chod aplikace

Aplikace potřebuje ke svému běhu, aby byla na počítači nainstalována podpora pro běh aplikací vytvořených v Microsoft .NET Framework 3.5. Na většině počítačů bývá tato podpora již nainstalována, protože je základní součástí operačního systému Windows.

### 7.2 Nastavení nezbytná pro správný chod webové služby

Na serveru, kde bude umístěna webová služba musí být nainstalována Internetová Informační Služba s podporou pro běh webových aplikací vytvořených v Microsoft .NET Framework 3.5.

U webové služby je třeba přidat nastavení do souboru „web.config“. V části souboru „appSettings“ se přidává klíč „FileDbComparatorWSAccessKey“ s přístupovým klíčem k webové službě a klíč „FileDbComparatorWSLogFile“ s cestou k souboru pro uchování záznamů o vzniklých výjimkách (Obr. 13).

```
<appSettings>
  <add key="FileDbComparatorWSAccessKey" value="h1B2Ngjg32vHHd673UVkb06vFShz7513"/>
  <add key="FileDbComparatorWSLogFile" value="log.txt"/>
</appSettings>
```

Obr. 13. Nastavení souboru „web.config“

Webová služba také vyžaduje, aby měla právo zápisu do souboru nastaveného k uchování záznamů o výjimkách. Také musí mít právo číst adresáře a soubory, které má porovnávat.

### 7.3 Porovnání souborů

Aby bylo možné spustit porovnávání souborů, musí mít uživatel vytvořený souborový komparátor. Pokud jej nemá, vytvoří ho pomocí tlačítka „Nový souborový komparátor“ na hlavním okně. V okně, které se objeví, zadá potřebné údaje a potvrdí tlačítkem „OK“. Poté může tento souborový komparátor spustit tak, že ho vybere ze seznamu „Komparátory:“ – „Souborové“ na hlavním okně a zmáčkne tlačítko „Porovnat“. Pokud jsou všechny údaje v komparátoru zadány správně, otevře se nové okno, ve kterém se načtou a zobrazí výsledky porovnání souborů. Z tohoto okna je také možné si prohlédnout nalezené rozdílné soubory. Stačí si vybrat požadovaný soubor a zmáchnout

tlačítko „Zobrazit“. Poté se objeví nové okno, kde je vedle sebe soubor v lokálním počítači a na serveru.

#### **7.4 Porovnání objektů v relační databázi**

Aby bylo možné spustit porovnávání objektů v relační databázi, musí mít uživatel vytvořený databázový komparátor. Pokud jej nemá, vytvoří ho pomocí tlačítka „Nový databázový komparátor“ na hlavním okně. V okně, které se objeví, zadá potřebné údaje a potvrdí tlačítkem „OK“. Poté může tento komparátor spustit tak, že ho vybere ze seznamu „Komparátory:“ – „Databázové“ na hlavním okně a zmáčkne tlačítko „Porovnat“. Pokud jsou všechny údaje v komparátoru zadány správně, otevře se nové okno, ve kterém se načtou a zobrazí výsledky porovnání objektů v relační databázi. Z tohoto okna je také možné si prohlédnout nalezené rozdílné objekty v relační databázi. Stačí si vybrat požadovaný objekt a zmáčknout tlačítko „Zobrazit“. Poté se objeví nové okno, kde je vedle sebe objekt v lokální databázi a v databázi na serveru.

## ZÁVĚR

Cílem této bakalářské práce bylo vytvořit aplikaci, která bude porovnávat soubory a objekty v relační databázi na serveru se soubory a objekty v relační databázi v lokálním počítači. Pro tuto aplikaci jsem navrhnul algoritmy pro porovnání souborů a objektů v relační databázi na serveru se soubory a objekty v relační databázi v lokálním počítači. Také jsem navrhnul grafické uživatelské rozhraní této aplikace. Všechny tyto návrhy jsem poté softwarově realizoval a to na platformě Microsoft .NET Framework 3.5.

Vytvořenou aplikaci jsem také otestoval proti živému serveru se soubory a živé relační databázi na serveru. Při všech testech aplikace fungovala správně a byla stabilní.

I přesto, že aplikace obsahuje všechny potřebné funkce a je plně funkční, je zde stále prostor pro její zlepšování. Například prohlížení souborů nebo objektů v relační databázi by bylo vhodné doplnit o inteligentní zvýrazňování rozdílů. Také u prohlížení tabulky v relační databázi by mohla aplikace zobrazovat místo sady tabulek s parametry dané tabulky raději skript, který by dokázal danou tabulku vytvořit. Přenášení dat ze serveru nebo serverové databáze by mohlo být rozšířeno o komunikaci po zabezpečeném kanále, což by zamezilo odposlechnutí přenášených a tím zvýšilo bezpečnost aplikace. Toto zabezpečení by mohlo být realizováno například pomocí protokolu SSL.



## ZÁVĚR V ANGLIČTINĚ

The main objective of this bachelor work was to develop an application that will compare files and objects in a relational database on a server with files and objects in a relational database on a local computer. For this application, I proposed algorithms for comparing files and objects in a relational database on a server with files and objects in a relational database on a local computer. I also designed graphical user interface of this application. I realized this application on platform Microsoft. NET Framework 3.5.

I also tested this application against a server with files and a relational database. In all tests application worked properly and was stable.

Even though the application contains all the necessary functions and is fully functional, there is still scope for its improvement. In viewing of the files or objects in a relational database, it would be appropriate to add the intelligent highlighting of differences. In viewing of the table in a relational database, the application should show script that could create the table instead of a set of tables with the parameters of the table. Transferring data from a server or database server could be expanded to communicate over secure channel, which would prevent the interception of transmitted data and thus increase safety of the application. This security could be implemented by SSL protocol.

**SEZNAM POUŽITÉ LITERATURY**

- [1] TROELSEN, Andrew. C Sharp a .NET 2.0 Profesionálně. Brno : Zoner press, 2006. 1200 s. ISBN 80-86815-42-0.
- [2] PETZOLD, Charles. Programování Microsoft Windows Forms v jazyce C Sharp. Brno : Computer Press, 2006. 360 s. ISBN 80-251-1058-3.
- [3] MACDONALD, Matthew. ASP.NET 2.0 a C Sharp : tvorba dynamických stránek profesionálně. Brno : Zoner press, 2006. 1376 s. ISBN 80-86815-38-2.
- [4] RFC 1321 - The MD5 Message-Digest Algorithm [online]. Dostupné z: <http://www.faqs.org/rfcs/rfc1321.html>.
- [5] The DES encryption algorithm [online]. Dostupné z: <http://www.iusmentis.com/technology/encryption/des/>.
- [6] BRUST, Andrew J. Mistrovství v programování SQL Serveru 2005. Brno : Computer Press, 2007. 847 s. ISBN 978-80-251-1607-4.
- [7] NAGEL, Christian, EVJEN, Bill, GLYNN, Jay, WATSON, Karli, SKINNER, Morgan. C Sharp 2008 Programujeme profesionálně. Brno : Computer Press, 2009. 1904 s. ISBN 978-80-251-2401-7.
- [8] SELLS, Chris. C Sharp a WinForms Programování formulářů Windows. Brno : Zoner press, 2005. 645 s. ISBN 80-86815-25-0.
- [9] WALTERS, Robert E., COLES, Michael, RAE, Robert, FERRACCHIATI, Fabio, FARMER, Donald. Mistrovství v Microsoft SQL Server 2008. Brno : Computer Press, 2009. 864 s. ISBN 978-80-251-2329-4.

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

API	Application Programming Interface.
CLI	Common Language Infrastructure.
CLR	Common Language Runtime
CSS	Cascading Style Sheets.
DES	Data Encryption Standard.
FTP	File Transfer Protocol.
FTPS	File Transfer Protocol Secure.
GUI	Graphical User Interface.
HTML	HyperText Markup Language.
HTTP	HyperText Transfer Protocol.
HTTPS	HyperText Transfer Protocol Secure.
IIS	Internetová Informační Služba.
JIT	Just-In-Time.
MD5	Message-Digest algorithm 5.
MS	Microsoft.
NNTP	Network News Transfer Protocol.
OLAP	Online Analytical Processing.
SEQUEL	Structured English Query Language.
SMTP	Simple Mail Transfer Protocol.
SOAP	Simple Object Access Protocol.
SQL	Structured Query Language.
SSL	Secure Sockets Layer.
T-SQL	Transact-SQL.
WSDL	Web Services Description Language.

XHTML Extensible Hypertext Markup Language.

XML Extensible Markup Language.

XSLT Extensible Stylesheet Language Transformations.

**SEZNAM OBRÁZKŮ**

Obr. 1. Hlavní okno aplikace .....	24
Obr. 2. Menu „Soubor“ hlavního okna .....	25
Obr. 3. Menu „Komparátor“ hlavního okna .....	25
Obr. 4. Okno pro vytváření a úpravu databázových komparátorů .....	26
Obr. 5. Okno pro vytváření a úpravu souborových komparátorů .....	26
Obr. 6. Okno výsledků porovnání pro souborový komparátor .....	27
Obr. 7. Okno výsledků porovnání pro databázový komparátor .....	27
Obr. 8. Menu „Soubor“ okna výsledků porovnání komparátoru .....	28
Obr. 9. Menu „Nastavení“ okna výsledků porovnání komparátoru.....	28
Obr. 10. Okno zobrazující soubor.....	29
Obr. 11. Okno zobrazující proceduru v relační databázi .....	29
Obr. 12. Okno zobrazující informace o tabulce v relační databázi.....	29
Obr. 13. Nastavení souboru „web.config“ .....	30

**SEZNAM TABULEK**

Tab. 1. Funkce tlačítek hlavního okna .....	24
Tab. 2. Funkce tlačítek okna výsledků porovnání komparátoru .....	27